



**Escola Universitària Politècnica
de Vilanova i la Geltrú**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

“INTRODUCCIÓN AL MICRO’PIC TRAINER PLUS”

AUTORES: Senén Iglesias Amado
Marc Tena Davós

TITULACIÓN: INGENIERIA TÉCNICA EN ELECTRÓNICA INDUSTRIAL

DIRECTOR: PERE PONS ASENSIO

DEPARTAMENTO: ESAII

FECHA: 29 de Maig de 2002

RESUMEN

El proyecto está dividido en cuatro partes.

En la primera se hace una introducción genérica a la MicroPic Trainer así como también se introducen los conocimientos necesarios de la MicroPic Trainer PLUS, tanto a nivel tecnológico como de arquitectura, con la función de que más adelante se puedan introducir nuevos conceptos sobre funcionamiento, programación y en consecuencia aplicaciones realizables sobre dicha placa.

La segunda parte trata del apartado de comunicaciones, tanto externas como internas, que contiene la placa. Además se comentará la existencia del PIC-BUS, que es el nexo de unión entre la MicroPic Trainer y la PLUS. La comunicación interna se realizará mediante el bus I²C y la externa mediante comunicación serie, vía canal RS-232. Se hará una descripción a nivel tecnológico, así como también del protocolo a seguir.

En la tercera parte hablaremos de las necesidades que conlleva la programación y testeo de los PIC utilizando la Trainer, tanto a nivel físico como a nivel de software. También se explica el porqué de la elección del software de programación escogido.

En la cuarta parte se hace una breve descripción de los componentes que intervienen en la programación del PIC, también se explica paso a paso el proceso a seguir y para concluir se analizan los ejemplos probados en clase para crear un enlace o nexo de unión entre la arquitectura de la MicroPic Trainer PLUS y sus aplicaciones.

Palabras clave

Microcontrolador PIC	Arquitectura	Interfaces	Comunicación
Teclado Hexadecimal	Modulo Visualización	E/S Paralelo	PIC-BUS
Canal série	Bus I ² C	Ensamblador	Picme-TR

INDICE

0.- OBJETIVO

1. CAPÍTULO I: INTRODUCCIÓN

1.1.- MicroPic' Trainer

1.1.1.- Introducción

1.1.2.-Arquitectura (Interfaces, módulos...)

1.2.- MicroPic' Trainer PLUS

1.2.1.- Introducción

1.2.2.-Arquitectura (Interfaces, módulos...)

- Teclado
- Módulo de visualización
- Módulo ADC-DAC
- Reloj-RAM
- E/S Paralelo

1.2.3.- Descripción tecnológica

1.2.4.- Componentes

2.- CAPÍTULO II: COMUNICACIONES

2.1.- La comunicación INTERNA: El BUS I²C y el PIC-BUS

2.1.1.-El PIC-BUS

2.2.2.- El BUS I²C

- Concepto de bus
- Terminología
- Características generales
- Transferencia de bit

2.2.- La comunicación EXTERNA: El canal serie RS-232

3.- CAPÍTULO III: IMPLEMENTACIÓN Y RECURSOS FÍSICOS

3.1.- Listado del material necesario.

3.2.- Discusión del software a utilizar.

4.- CAPÍTULO IV: PROGRAMACIÓN CON EL μ PIC-TRAINER PLUS

4.1.- El ensamblado

- El MPASM

4.2.- La carga de datos

- El PICME-TR

4.3.- Las librerías

4.4.- Aplicaciones

- Descripción del Proceso
- Enlace entre arquitectura y aplicaciones

5.- BIBLIOGRAFÍA

0.- OBJETIVO

Los objetivos que se trataran de alcanzar en la realización del proyecto serán los siguientes:

A)- Conocimientos básicos sobre el “MicroPic’ Trainer”.

- Características generales.

B)- Conocimiento en profundidad del “MicroPic’ Trainer PLUS”.

- Características generales.
- Tipo de arquitecturas.
- Características técnicas (Buses, memorias y registros, convertidores,...)
- Conocimientos básicos del lenguaje de programación, tomando como base aplicaciones ya estudiadas.

1.- CAPÍTULO I: INTRODUCCIÓN

1.1.- MICROPIC’ TRAINER

1.1.1.- Introducción

El sistema MicroPic Trainer consiste en un equipo completo para la evaluación de aplicaciones basadas en los microcontroladores PIC de gama media de MICROCHIP. Dispone de una serie de periféricos básicos de E/S con los que se puede verificar el funcionamiento de una aplicación, así como la circuiteria necesaria para la grabación de todos los modelos de microcontroladores PIC de gama media de 18 y 28 patitas.

Se denomina "sistema de desarrollo" el equipo físico y el conjunto de programas que conforman una herramienta capaz de desarrollar todas las operaciones necesarias para diseñar un proyecto basado, en un microcomputador programable, que al estar incorporado en un circuito integrado recibe el nombre de **microcontrolador**. En las siguientes líneas intentaremos hacer una breve descripción del esquema electrónico y el funcionamiento de este sistema.

- Las características del sistema son las siguientes:

- 1.- Alimentación desde un transformador de 12 VAC o desde dos pilas de 9 VDC. El circuito de estabilización esta incluido en la placa. Con el se obtiene la tensión general de +5 V y la de grabación de +13.8 V.
- 2.- Soporta cualquier microcontrolador de gama media tanto de 18 como de 28 patitas. La frecuencia de trabajo por defecto es de 4 MHz.
- 3.- Generación automática de RESET y manual mediante pulsador.
- 4.- 5 Entradas digitales implementadas mediante 5 conmutadores.
- 5.- 4 Entradas de tensión analógica, variables mediante potenciómetros.
- 6.- 8 Salidas digitales conectadas a 8 indicadores luminosos tipo LED y/o a display de 7 segmentos.
- 7.- Conexión de un modulo de visualización tipo LCD de 2*16 (dos líneas de 16 caracteres cada una) y control de contraste mediante potenciómetro.
- 8.- Tanto los periféricos de entrada como los de salida se pueden desactivar o no mediante sus respectivos jumpers.

9.- Conector de expansión PIC-BUS donde todas las señales de E/S del microcontrolador, están disponibles para ser empleadas por otros periféricos distintos a los que dispone MicroPIC Trainer.

10.- La señal RA4/TOCKI se puede emplear como entrada o salida digital o entrada de pulsos de reloj para el Timer 0, desde el exterior.

11.- La señal RB0/INT se puede emplear como entrada o salida digital o entrada de interrupción externa.

12.- Circuito grabador "en circuito". Permite, con el software de control necesario, grabar el PIC que en ese momento este insertado en cualquiera de los zócalos. Es especialmente útil para el modelo 16C84, que al disponer de memoria EEPROM de programa se puede grabar y regrabar tantas veces como sea necesario sin necesidad de retirarlo del zócalo. Lo mismo sucede con el 16F84, con memoria FLASH.

13.- Led indicador de presencia de la Vpp de grabación.

14.- Conector DB25 para conexión con el canal paralelo de cualquier ordenador personal.

15.- Software en castellano para realizar todas las operaciones propias de grabación: Lectura, grabación, verificación, borrado, edición, etc.

16.- Todos los componentes que forman MicroPic Trainer van sobre una placa de circuito impreso de tipo profesional. El conjunto se suministra completamente montado o en forma de Kit para que sea el propio usuario quien lo ensamble.

1.1.2.- Arquitectura

LA FUENTE DE ALIMENTACIÓN

Es la encargada de obtener las tensiones estabilizadas de trabajo: alimentación general de +5V y la tensión de grabación de +13.8V.

La tensión de entrada se aplica mediante 2 pilas de 9V o bien desde un transformador de 12VAC conectado a J1. En este caso la tensión alterna se rectifica mediante el puente de diodos D1 y se filtra mediante los condensadores C1 y C2, para obtener una continua sin estabilizar.

Sea como fuere, la tensión continua así obtenida se estabiliza a +5V mediante el regulador U6, -uA7805, y +13.8V mediante el regulador uA7812 junto con los tres diodos D2-D4, 1N4007.

EL MICROCONTROLADOR

El sistema MicroPIC Trainer esta diseñado para trabajar con cualquier modelo de microcontrolador PIC de la gama media encapsulado con 18 o 28 patitas. Ésta es la razón de la existencia de dos zócalos para soportar ambos tipos.

Las diferencias básicas entre los modelos de 18 y 28 patitas consiste en que los primeros disponen de una puerta A de 5 líneas (RA0-RA4) y una puerta B de 8 (RB0-RB7), y una puerta C de 8 líneas (RC0-RC7).

En MicroPIC Trainer se emplean las 5 líneas de la Puerta A y las 8 de la Puerta B, que son comunes a los dos modelos de PIC. En cualquier caso, las líneas de estas puertas, junto con las de la puerta C, están disponibles para el usuario a través del conector PIC-BUS de expansión.

La frecuencia de trabajo para cualquiera de los dos tipos de PIC queda establecida mediante el cristal de cuarzo X1 y los condensadores C6 y C7. Esta es, por defecto de 4MHz, aunque el usuario puede modificarla cambiando los valores del cristal y los condensadores según las características técnicas del modelo de PIC elegido.

ENTRADAS DIGITALES

Están formadas por 5 interruptores (SW2-SW6) conectados a las líneas RA0-RA4 de la puerta A, capaces de introducir niveles lógicos “1” y “0” por esas líneas. Dichas líneas pueden programarse para actuar de diferentes formas. Así, RA0-RA3 pueden actuar, además, como entradas analógicas y RA4 como entrada exterior de pulsos de reloj para el TMR0 (T0CKI).

Estas líneas tienen asociados unos jumpers con los que se selecciona el tipo de entrada que se les va a aplicar. Los jumpers J8(0)-J8(3) seleccionan individualmente si por RA0-RA3 se van a introducir entradas digitales con los interruptores SW2-SW5, entradas analógicas con los potenciómetros P2-P5 o, simplemente, si estas líneas van a quedar desconectadas tanto de los interruptores como de los potenciómetros y poder así usarse con otros periféricos distintos.

Con el jumper J9 se selecciona si RA4 actúa como entrada digital procedente del interruptor SW6, entrada de señal de reloj para el TMR0 (T0CKI), que se aplica desde el exterior mediante el conector J2(1), o simplemente, si esta línea va a quedar desconectada y libre, poder usarla con otro periférico distinto.

ENTRADAS ANALÓGICAS

Están conectadas a la puerta B y consisten, por una parte, en una barra de diodos leds luminosos (D8) que representan el estado lógico de las señales RB0-RB7, y por otra, un display alfanumérico de 7 segmentos.

Estos periféricos están conectados entre sí en paralelo a la puerta B. Para evitar un exceso de consumo en dicha puerta, tanto los leds como el display disponen de los jumpers J7 y J5, respectivamente, con los que se les puede desconectar si la aplicación en curso no hace uso de cualquiera de ellos.

El pack de resistencias RPACK 1 contiene 8 resistencias de 330 Ohms que actúan como resistencias de absorción o limitación.

La barra de diodos luminosos D8 contiene 10 diodos tipo led. Los 8 primeros, S0-S7, representan, en caso de estar habilitados (J7), el estado de las señales RB0-RB7; el noveno led no se emplea y el décimo sirve para indicar la presencia de tensión de +5V (Vcc ON).

La línea RB0 puede actuar como entrada de interrupción externa INT. Dicha señal de entrada se aplica por el conector J2(3). Con el jumper J10 la señal RB0 se conecta a los leds y al display, a la entrada de interrupción INT, o simplemente, se deja sin conexión.

Los periféricos conectados a la puerta B pueden deshabilitarse mediante sus respectivos jumpers. De esta forma no tienen efecto alguna sobre esa puerta, por lo que las señales RB0-RB7 disponibles en el conector de expansión PIC-BUS pueden ser empleadas por otro tipo de periféricos en otras configuraciones.

EL MÓDULO LCD

Se trata de un módulo de visualización alfanumérico de cristal líquido, capaz de presentar 2 líneas con 16 caracteres cada una.

Aplicándole los códigos necesarios se puede provocar diferentes efectos de visualización como parpadeo, scroll, activación de un cursor, etc. Incluso es posible la generación de nuevos caracteres definidos por el usuario.

Las líneas de datos D0-D7 están conectadas con las 8 líneas de la puerta B (RB0-RB7).

Esta puerta, a veces, actúa como salida del PIC y entrada hacia el módulo. Por ella se le aplican los diferentes códigos de control para realizar diferentes efectos de visualización, así como los códigos ASCII de los caracteres a visualizar. En otras ocasiones la puerta B debe

actuar como entrada hacia el PIC, ya que a su través el módulo LCD devuelve códigos indicando su estado interno, el contenido del buffer de memoria interna, etc.

El módulo está conectado a las líneas RA0, RA1 y RA2 de la puerta A del PIC. Estas líneas actúan como salida y se emplean para enviar las siguientes señales de control al módulo:

- RS=0** → Selección del registro de instrucciones.
- RS=1** → Selección del registro de datos.
- R/W=0** → Ciclo de escritura sobre el módulo LCD.
- R/W=1** → Ciclo de lectura del módulo LCD.
- E=0** → Módulo LCD desactivado.
- E=1** → Módulo LCD activado.

Hay que destacar que la señal E está conectada con RA2 a través de jumper J6. Si dicho jumper está cerrado, RA2 controlará la habilitación o no del módulo LCD. Por el contrario, si estuviera abierto, la señal E queda conectada a tierra (nivel “0”) a través de R16. De esta forma el módulo estará siempre desconectado y las líneas tanto de la puerta A (RA0-RA2) como de la B (RB0-RB7) pueden ser empleadas por otros periféricos, ya sean los que están disponibles en μ PIC Trainer o cualquier otro conectado al conector de expansión PIC-BUS.

El módulo LCD es un periférico de visualización muy utilizado en aplicaciones reales.

Permite una cómoda, atractiva y versátil forma de visualización alfanumérica e incluso gráfico.

EL CIRCUITO GRABADOR

Se trata de una de las secciones más interesantes del sistema μ PIC Trainer. Su importancia radica en la posibilidad de realizar la grabación del PIC “en circuitos”. Esto significa que el PIC va a poder ser manipulado (leído, grabado, borrado, etc.) directamente alojado en el zócalo de trabajo.

Esta ventaja es aún más relevante si se emplea el modelo de PIC16C84. Este modelo dispone de memoria EEPROM como memoria de programa. Esta memoria, a pesar de no ser volátil, puede grabarse y borrarse casi un número indefinido de veces. Así pues, este PIC, insertado en su correspondiente zócalo, va a poder ser grabado con múltiples programas y aplicaciones, al tiempo que directamente se van ejecutando y evaluando sobre μ PIC Trainer. Lo mismo pasa con el PIC16F84.

El circuito grabador permite grabar todos los modelos de 18 y 28 patitas de la gama media de PIC. Mediante el conector J3 se conecta al canal paralelo de cualquier ordenador personal tipo PC/XT/AT. El programa de control PICME-TR, totalmente en castellano e incluido en el kit μ PIC Trainer, se encarga de gestionar todas las operaciones, tales como leer el contenido del PIC (si no está protegido), grabar, verificar, borrar (Sólo el 16X84) establecer la palabra de configuración, etc.

La conexión entre el PC y μ PIC Trainer se realiza a través del canal paralelo. El circuito integrado U3-CONTROL#4.0 se encarga de verificar el hardware de μ PIC Trainer y la comunicación entre éste y el PC.

La lectura/grabación de los PIC de la gama media se realiza en serie. Los bits de datos se leen o escriben secuencialmente, uno tras otro, a través de la señal RB7 y son aplicados al canal paralelo mediante los amplificadores U4D y U4F, para su posterior tratamiento por el software de control.

La señal RB6 actúa como entrada de reloj para sincronizar la transferencia de los bits de datos durante el proceso de lectura/grabación del PIC. Esta señal de reloj es generada por el software de control y se aplica por el canal paralelo a través del amplificador U4C.

El software de control también genera una señal que, mediante U4B, se aplica al transistor de conmutación Q1. Este transistor al activarse envía los 13.8V generados desde la fuente de alimentación a la patilla MCLR/VPP del PIC que se está leyendo o grabando. Al mismo tiempo el diodo led D5 se ilumina indicando actividad en el circuito grabador de μ PIC Trainer. Finalmente, a través de U4A, el software de control genera la señal MCLR, que resetea y reinicia al microcontrolador. Normalmente esto ocurre al finalizar cualquier operación de lectura y/o grabación, con lo que el PIC se lanza inmediatamente a ejecutar el programa que contenga en su memoria. Cabe indicar que, en cualquier caso y en cualquier momento, el usuario puede provocar el Reset manualmente accionando el pulsador RESET SW1.

Dado que las señales que se emplean para la lectura/grabación de un PIC son RB6 y RB7 y éstas a su vez están conectadas a los leds, display y módulo LCD, es importante desconectar estos periféricos mediante los jumpers J7, J5 y J6, respectivamente, para que no supongan carga alguna a dichas señales y alteren la información leída o grabada.

EL CONECTOR DE EXPANSIÓN

Se trata de un conector de 26 contactos en el que están disponibles todas las señales de las diferentes puertas de E/S, además de las de alimentación y MCLR. Esto permite conectar futuras expansiones desarrolladas por **Microsystems Engineering** o por el propio usuario, manejando nuevos periféricos. Esta posibilidad tiene especial interés en la conexión de la placa de prototipos PIC18-ME, donde pueden alojarse los periféricos de la aplicación, mientras que se depura el software desde el sistema de desarrollo.

Para que no haya conflicto entre los periféricos conectados al PIC-BUS y los ya existentes en μ PIC Trainer, es necesario desconectar, mediante sus correspondientes jumpers, aquellos que vayan a ser reemplazados.

1.2.- MICROPIC' TRAINER PLUS

1.2.1.- Introducción

La MicroPic' Trainer Plus es un equipo que conectado al sistema de desarrollo MicroPic'Trainer permite ampliar y experimentar prácticamente todos los recursos más avanzados e interesantes que soportan los microcontroladores PIC.

Si vemos los diferentes modelos de PIC's nos encontramos con dispositivos que implementan, mediante hardware, distintos módulos que permiten la comunicación con dispositivos externos. Como los módulos SCI y SSP.

- El módulo SCI (Interface de Comunicación Serie) hace posible la comunicación serie entre el microcontrolador y cualquier otro periférico externo o cualquier otro periférico cuyo interfaz sea del tipo RS-232, RS-485, etc..
- El módulo SSP (Puerta Serie Síncrona) está pensado para la conexión serie entre el PIC y ciertos dispositivos integrados diseñados al efecto. Abundan las memorias RAM y EEPROM.
 - La I²C consiste en una comunicación síncrona a 3 hilos con distintos dispositivos diseñados al efecto. Abundan las memorias RAM y EEPROM, se pueden encontrar convertidores ADC y DAC, relojes en tiempo real, etc...

CARACTERÍSTICAS GENERALES

Las características más relevantes del sistema de ampliación MicroPIC Trainer Plus son:

- Conexión directa mediante cable plano de 26 hilos al conector de expansión "PIC-BUS" de microPIC Trainer.
- Alimentación única de +5Vcc que se toma directamente del PIC-BUS o bien se aplica a través de las bornas de conexión exterior.
- Tamaño de la placa 100*160 mm Eurocard.
- Teclado numérico hexadecimal de 4*4 conectado a las líneas RB0-RB7 de la puerta B.
- Canal de comunicaciones serie RS-232 con conector estándar DB9 macho. Este canal se puede gestionar mediante las líneas RC6 y RC7 para aquellos dispositivos que implementan la USART por hardware, o bien mediante las líneas RB4 y RB5 para la gestión por software. Se selecciona mediante Jumpers.
- Interfaz I²C gestionado por las líneas RC3-RC4 para los dispositivos que lo implementan por hardware, o bien mediante las líneas RB6-RB7 para la gestión software. Se selecciona por jumpers.
- Display de 4 dígitos de 7 segmentos gestionado por el dispositivo I²C SAA1064.
- Cuatro canales de conversión analógico-digital (ADC) y un digital-analógico (DAC) proporcionados por el dispositivo I²C PCF8591. Tanto los 4 canales AD como el DA están disponibles en el exterior mediante las bornas de conexión para ser usados por los distintos transductores.

La tensión de referencia empleada por los convertidores ADC y DAC se puede también aplicar desde el exterior mediante la borna correspondiente.

- Ocho líneas de E/S digital proporcionadas por el dispositivo I²C PCF8574. Estas líneas están disponibles en las bornas de conexión para ser empleadas por los periféricos que se deseen controlar.

Cada línea está asociada a un diodo led activado por nivel "0" para pilotar el estado lógico de esas 8 señales.

- Reloj/calendario en tiempo real controlado por el dispositivo I²C PCF8583. Además de las clásicas funciones horarias dispone de alarma y puede actuar como

temporizador/contador. Está alimentado por batería de Ni/Cd recargable que mantiene al circuito en hora aún a falta de tensión de alimentación.

El circuito posee también 240 bytes de memoria RAM no volátil.

- Va acompañado de un manual con instrucciones de montaje, funcionamiento, experimentos prácticos e información técnica, así como un disco de 3.5" con software de aplicación y ejemplos.

1.2.2.-Arquitectura del Micro'Pic Trainer Plus

A continuación se explicarán algunas de las diferentes secciones que configuran el microPIC Trainer Plus.

EL TECLADO.

Mediante el conector CN1 de 26 contactos, se conecta el cable plano a la placa microPIC Trainer, de donde se obtienen todas las señales del PIC que en ella esté insertado (Puerta A, Puerta B, Puerta C, Alimentación y MCLR).

Las 8 líneas de la puerta B (RB0-RB7) se conectan directamente al teclado de 4*4:

- RB7-RB4 se configurarán como salidas para ir activando secuencialmente cada una de las filas.
- RB3-RB0 se configurarán como entradas desde las columnas del teclado y son leídas para determinar si hay alguna tecla pulsada.
- RB6 y RB7 también pueden actuar como señales SDA y SCL para el interfaz I²C en caso de gestionarse por software. Si se está trabajando con un PIC que incorpora hardware I²C son las líneas RC4 y RC3 las que envían las señales SDA y SCL, respectivamente. Para seleccionar el modo de implementar el bus I²C, lo haremos con los jumpers J1 y J2 que activan las dos señales.

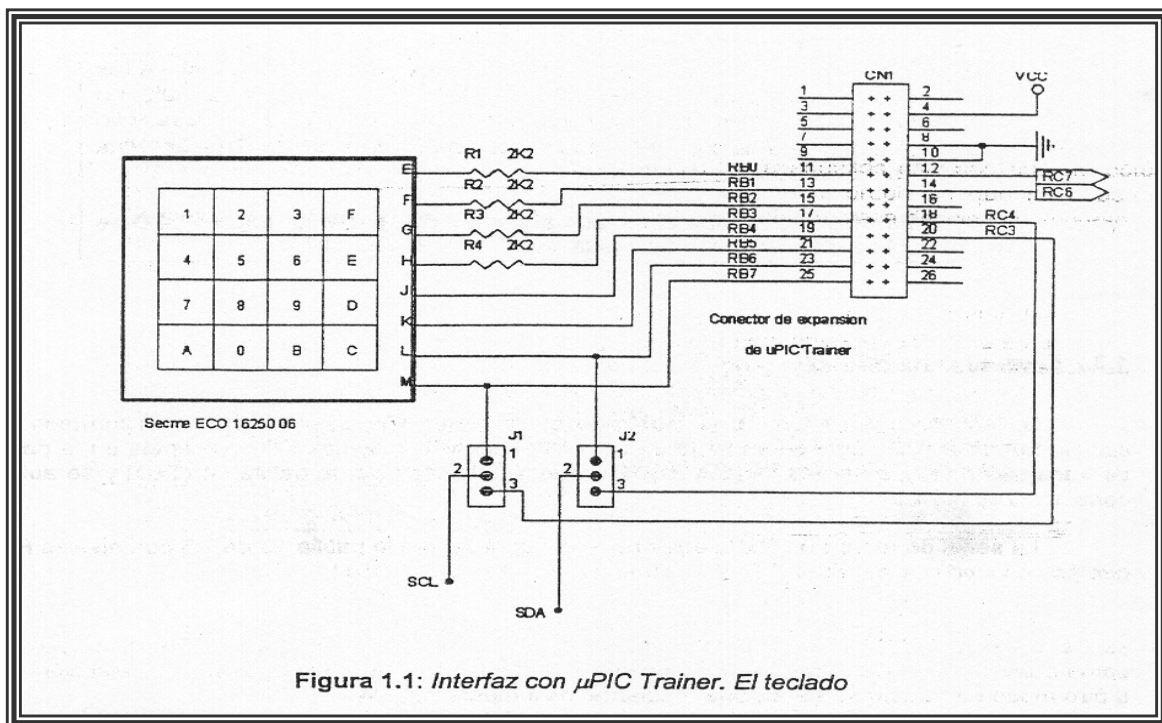


Figura 1.1: Interfaz con μ PIC Trainer. El teclado

EL MÓDULO DE VISUALIZACIÓN.

Consiste en un conjunto de 4 displays de 7 segmentos controlados por el dispositivo I²C SAA1064 (U1).

El MASTER envía la información mediante las líneas SCL y SDA del bus I²C, que observaremos sobre los 4 displays. La dirección I²C del dispositivo es de 7 bits y los 5 bits de más peso están establecidos por el fabricante con la combinación 01110. Los otros 2 bits los establece el usuario mediante la tensión que se aplique a la patilla 1 (ADR):

Tensión en ADR	Bits
GND	00
3/8 de Vcc	01
5/8 de Vcc	10
Vcc	11

Por tanto, la dirección asignada en MicroPIC Trainer Plus es 0111011x (x=0 o x=1) según la tensión que se tenga.

Cuando el circuito SAA1064 recibe la información vía I²C, realiza un multiplexado de los 4 displays para visualizarla. Por las líneas P1-P8 aparece la información a representar por el dígito 1 y, por las líneas P9-P16 la del dígito 3. Al mismo tiempo aparece la señal MX1 activando el transistor Q1 que gobierna los ánodos comunes de dichos dígitos.

Pasado un tiempo, por las líneas P1-P8 y P9-P16 aparece la información para los dígitos 2 y 4 junto a la señal MX2 que activa al transistor Q2 gobernando los ánodos comunes de dichos dígitos. Este proceso se repite indefinidamente y a una cierta velocidad, ya que sólo hay dos dígitos activados en cada instante y da la sensación de que los cuatro están activados.

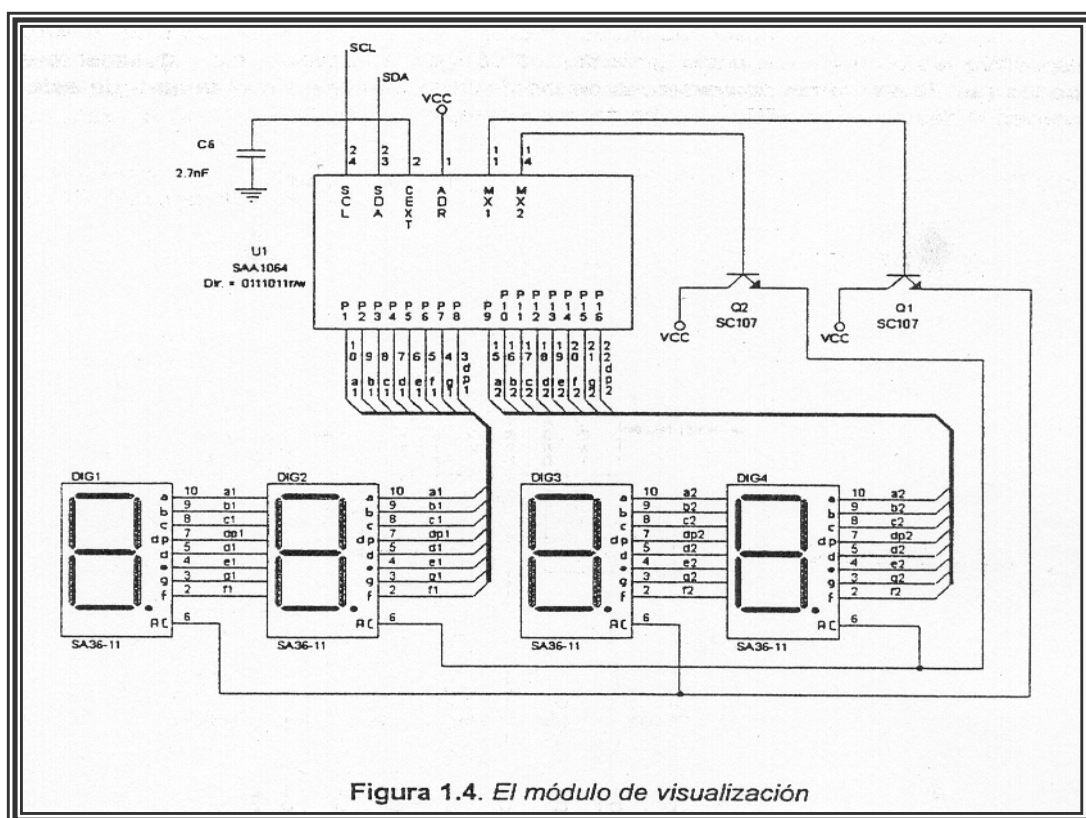


Figura 1.4. El módulo de visualización

EL MÓDULO ADC/DAC.

Se centra en el dispositivo I²C PCF8591 (U3). Su dirección está formada por 4 bits establecidos en fábrica con la combinación 1001 y otros 3 bits que dependen del estado lógico de las líneas A0, A1 y A2.

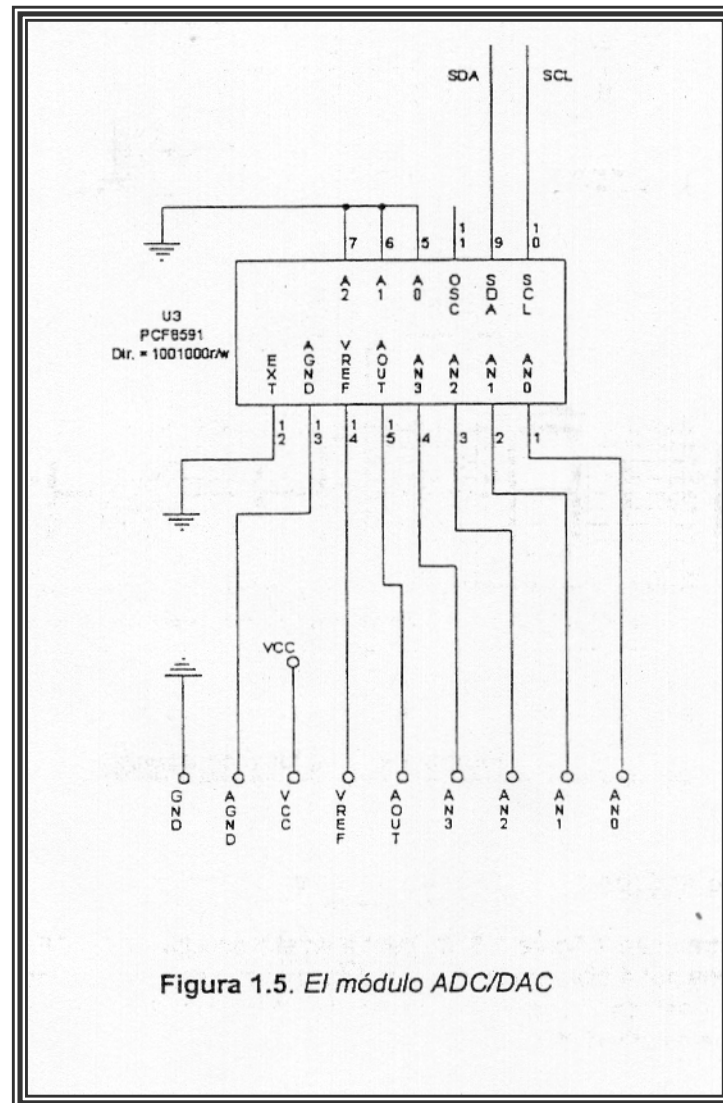
En el caso del microPIC Trainer Plus la dirección sería 1001000x.

El circuito consiste en un convertidor Analógico/Digital (ADC) de 8 bits y cuatro canales analógicos de entrada, donde las señales a convertir entran por AN0, AN1, AN2 y AN3.

También dispone de un convertidor Digital/Analógico (DAC) de 8 bits cuya tensión de salida se obtiene en AOUT.

La señal analógica de tierra y de referencia se aplican respectivamente a las líneas AGND y VREF.

El MASTER a través de las líneas SCL y SDA envían al dispositivo las órdenes de funcionamiento, los códigos a convertir, etc., y recibe los resultados de las conversiones de los distintos canales.



RELOJ/CALENDARIO + RAM.

Es un dispositivo que se adapta al bus I²C modelo PCF8583 (U2) que contiene un reloj/calendario en tiempo real. También realiza funciones de alarma y temporización. Dispone en su interior de 256 posiciones de memoria RAM estática, de las cuales las 16 primeras están reservadas para las funciones de reloj, calendario, alarma y temporización. Las 240 restantes son para que el usuario pueda guardar su información.

Las líneas SCL y SDA del bus I²C son con las que el dispositivo comunicará con el MASTER (el PIC empleado en la aplicación) para su lectura y/o escritura.

La dirección I²C del dispositivo es la 1010000x y viene determinada por la propia estructura interna del PCF8583P y por el estado lógico de la patilla 3 (A0) que en el PIC Trainer Plus es de nivel "0". La patilla 1 (OSCI) es la entrada de reloj, a la que se aplica la frecuencia patrón la cual mantiene al reloj/calendario en hora. Esta frecuencia puede ser de 50 Hz o la obtenida por un cristal de 32.768 Hz conectado entre esta patilla (OSCI) y la 2 (OSCO).

La patilla 7 (INT) es una señal de salida que se activa por nivel bajo en caso de alarma horaria o se alcance una determinada temporización. Se utilizar para activar cualquier periférico o provocar interrupciones al MASTER que controla a este dispositivo.

El circuito PCF8583P se puede alimentar mediante la tensión Vcc o bien mediante una batería recargable de Ni/Cd. En presencia de la tensión Vcc, el circuito se alimenta de ella a través del diodo D5. Al mismo tiempo la batería se carga a través de D5 y R16.

Ante falta de alimentación es la pila la que suministra la tensión al circuito y mantenga los datos que haya en su memoria RAM.

Se dispone así de un sistema de almacenamiento no volátil.

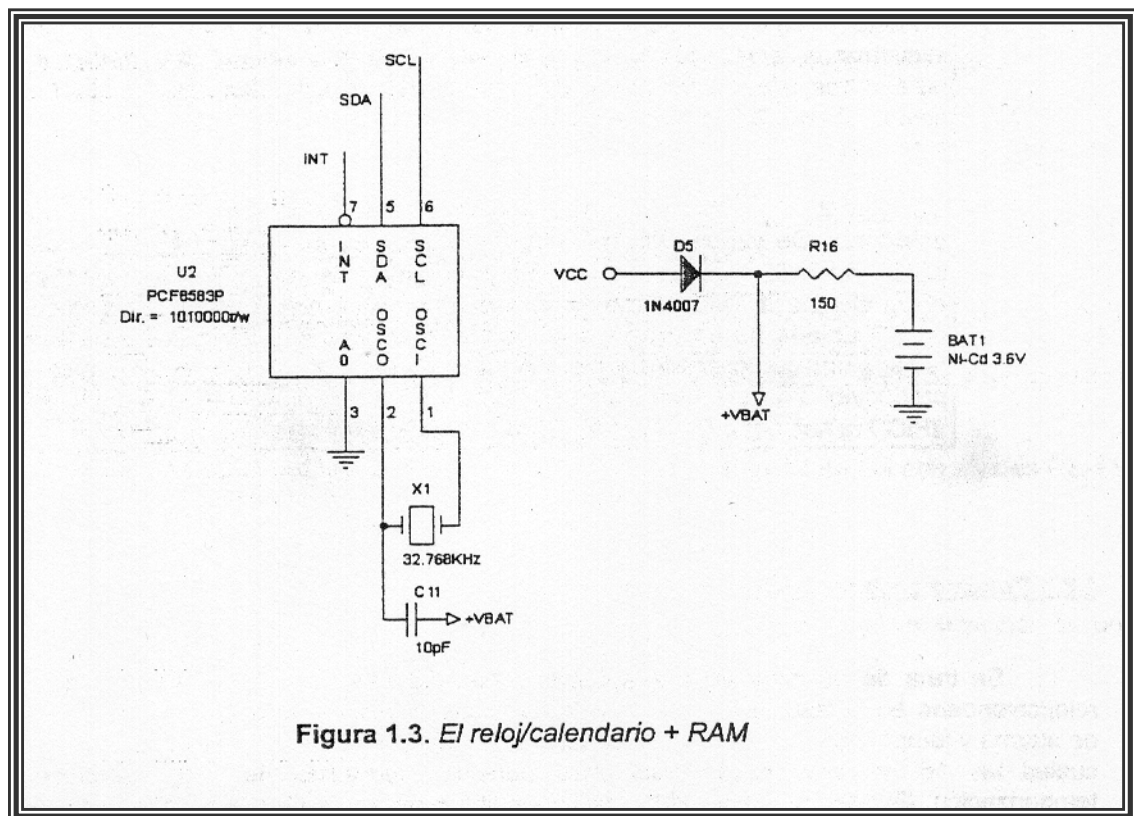


Figura 1.3. El reloj/calendario + RAM

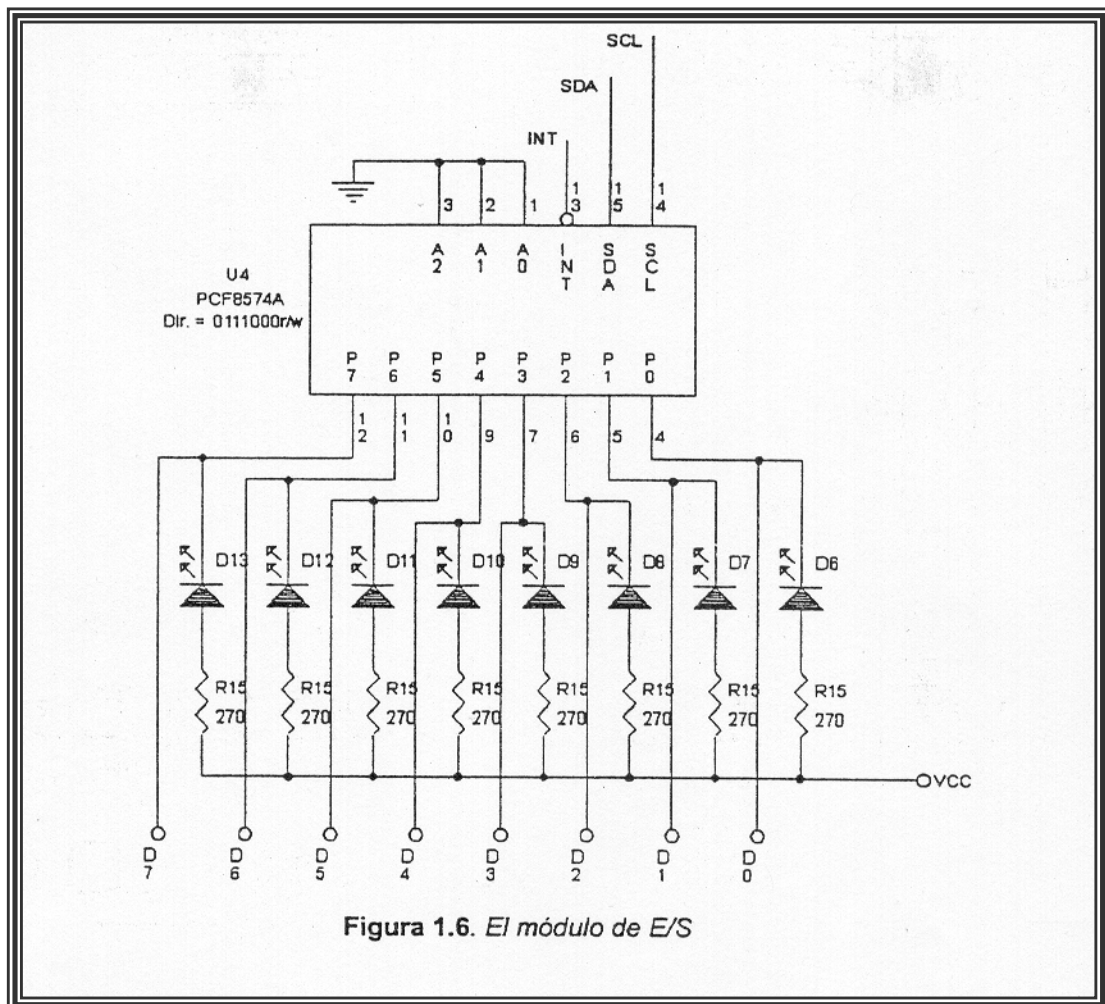
LAS E/S EN PARALELO.

Están basadas en el dispositivo I²C PCF8574A (U4). Es un circuito cuya dirección está formada por 4 bits cuya combinación es 0111 y el estado lógico de las líneas A0, A1 y A2 es la 0111000x.

Se trata de un circuito con 8 líneas que pueden actuar como entradas o salidas según se lean o escriban (P0-P7). A estas líneas se conectan los periféricos que queramos controlar a través de las correspondientes bornas de conexión. Además cada línea dispone de un led que pilota y representa su estado lógico, activándose éstos por lógica negativa.

A través de las líneas SCL y SDA el MASTER lee o escribe los valores binarios presentes en las líneas P0-P7. La señal de salida INT, activa por "0" se puede utilizar como interrupción al MASTER, que ocurre cada vez que una línea de entrada cambia de estado.

El circuito permite ampliar el número de líneas de E/S de un PIC.



ESQUEMA GENERAL.

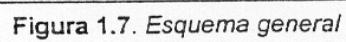


Figura 1.7. Esquema general

1.2.3.- Descripción tecnológica

DESCRIPCIÓN DEL PCF8583 RELOJ/CALENDARIO + SRAM.

Circuito CMOS de bajo consumo con 2048 bits de SRAM organizados en 256*8 bytes. Las direcciones y datos se transfieren en serie según el bus I²C. El registro interno de direcciones se incrementa automáticamente al leer o escribir un byte. Tiene una línea externa de direccionamiento por hardware (A0) que permite conectar dos circuitos iguales en el mismo bus.

Incluye un oscilador interno de 32.768 Hz empleado por un reloj/calendario en tiempo real. Los 8 primeros bytes de la RAM se utilizan para controlar las funciones del reloj/calendario y los 8 siguientes para las funciones de alarma.

DESCRIPCIÓN DEL SAA1064. CONTROLADOR DE DISPLAY.

Está diseñado para el control de hasta 4 displays d 7 segmentos tipo led y con punto decimal, mediante el multiplexado de dos pares de dígitos.

Con interface I²C, el circuito puede trabajar junto con otros tres conectados al mismo bus gracias a las cuatro direcciones posibles diferentes, seleccionadas por hardware.

DESCRIPCIÓN DEL PCF8591. CONVERTIDOR A/D Y D/A.

Es un circuito con cuatro canales analógicos de entrada, uno de salida, interface con el bus I²C y tensión de alimentación única. Dispone de tres líneas de direccionamiento por hardware (A0-A2) que permite conectar 8 chips idénticos sobre el mismo bus.

DESCRIPCIÓN DEL PCF 8574A, E/S DIGITALES.

Dispositivo CMOS de bajo consumo diseñado para interface con el bus I²C que proporciona una ampliación de 8 líneas de E/S digitales y que se puede utilizar en cualquier sistema microcontrolador con hardware y/o software para el I²C.

Incluye salidas "latcheadas" con amplificadores de alta corriente para activar directamente cargas tipo LED. Es capaz de provocar una interrupción al sistema cuando se produce una llamada sobre sus líneas de entrada. Actúa como dispositivo SLAVE del bus.

1.2.4.- Componentes

Referencia	Valor	Descripción
U1	SAA1064	Controlador de displays 7 segmentos
U2	PCF8583P	Reloj/calendario + RAM
U3	PCF8591	Convertidor AD y DA
U4	PCF8574A	E/S paralelo
U5	MAX232	Convertidor de niveles lógicos
DIG1-DIG4	SA36-11	4 displays de ánodo común
D5	1N4007	Diodo
D6-D13	Led	8 diodos led de 3 mm
Q1-Q2	SC 107	2 transistores NPN
R1-R7	2K2	7 resistencias de ¼ W
R8-R15	270	8 resistencias de ¼ W
R16	150	1 resistencia de ¼ W
R17	10K	1 resistencia de ¼ W
R18	100 K	1 resistencia de ¼ W
C1-C5	100 nF	5 condensadores
C6	2.7 nF	1 condensador
C7-C10	22 µF	4 condensadores electrolíticos radiales de 25V
C11	10 pF	1 condensador
BAT1	3.6 V	Batería recargable de Ni/Cd
X1	32,768 KHz	Cristal de cuarzo
CN1	26 c	Conector de 26 contactos para cable plano, macho, recto
CN2	DB9	Conector DB9 macho, codo de bajo perfil
J1-J4		4 tiras con 3 postes rectos cada una y sus correspondientes caperuzas.
Teclado		Teclado de 4 x 4 SECME mod. ECO 16250 06
Bornas		6 bornas de 3 contactos para circuito impreso
Zócalos		1 de 24, 3 de 16 y 1 de 8 pines
Separadores		4 de 6 mm y 4 de 10 mm
Tuercas		8 tuercas métrica 10
Cable		Latiguillo de cable plano de 26 hilos acabado en 2 conectores hembra.
Pines		Tira de 8 pines hembra
C. Impreso		Placa de C. Impreso de Microsystems Engineering
Disco		Disco de demostración con programas de ejemplo
Manual		Manual de usuario de µPIC Trainer Plus

2.- CAPÍTULO II: COMUNICACIONES

2.1.- La comunicación INTERNA: El BUS I²C y el PIC-BUS

2.1.1.-El PIC-BUS

Tal como se dijo en el capítulo 1, la MicroPic Trainer PLUS va conectada a la MicroPic Trainer a través de un conector de expansión formado por cable plano de 26 hilos. A través de este conector la MicroPic Trainer gestiona todas las E/S externas a la propia placa, y esto es aprovechado para conectar su ampliación: la MicroPic Trainer PLUS.

Cabe recordar que para que no haya conflicto entre los periféricos conectados al PIC-BUS y los ya existentes en μ PIC Trainer, es necesario desconectar, mediante sus correspondientes jumpers, aquellos que vayan a ser reemplazados.

2.2.2.- El BUS I²C

•Concepto del bus I²C.

Las líneas, SDA (datos) y SCL (reloj), transportan la información entre los diferentes dispositivos conectados al bus. Cada dispositivo se identifica por una única dirección y puede transmitir o recibir dependiendo de la operación que se vaya a realizar. Un controlador de LCD, por ejemplo, solo recibe mientras que una memoria de tipo RAM puede transmitir o recibir datos según se vaya a leer o escribir.

Los dispositivos pueden clasificarse en **MASTER** (principal) o **SLAVE** (secundario). El MASTER es el que indica la transferencia de datos y genera la señal de reloj. Los dispositivos direccionados por un MASTER, se considera un SLAVE.

El I2C es un bus **MULTIMASTER**. Puede haber más de un MASTER conectado y controlando el bus. Normalmente se tratan de microcontroladores o microcomputadores.

Dada la posibilidad de que existieran varios MASTER, podría ocurrir que más de uno inicie la transferencia al mismo tiempo. Para evitar el caos se ha desarrollado un sistema de arbitraje del bus. Consiste en una conexión tipo AND entre todos los dispositivos conectados al bus. Cuando uno o más MASTERS colocan información en la línea SDA, verifican si el bit que ellos sacan coincide con el nivel lógico de dicha línea. Si un MASTER saca un “1” pero la línea SDA está a “0” coincidiendo con un pulso de reloj, pierde la posesión del bus cancelando la transmisión. El nivel lógico “0” (bit dominante) presente en la línea SDA procederá de un MASTER distinto.

Las señales de reloj durante el arbitraje del bus son una combinación entre las señales de clock de los distintos MASTERS conectadas entre sí a la línea SCL mediante una conexión tipo AND.

•Terminología del bus I2C.

TRANSMISOR: Dispositivo que coloca información en la línea SDA.

RECEPTOR: Dispositivo que recibe información por la línea SDA.

MASTER: Dispositivo que inicia la transferencia, genera la señal de reloj y finaliza la transferencia.

SLAVE: Dispositivo seleccionado por el MASTER.

MULTI-MASTER: Cuando más de un MASTER puede hacerse con el control del bus sin corromper la información.

ARBITRAJE: Procedimiento que garantiza que, si más de un MASTER acceden simultáneamente al bus, únicamente uno de ellos se hace cargo del mismo con objeto de no alterar la información.

SINCRONIZACIÓN: Procedimiento para sincronizar la señal de reloj en un Sistema MULTI-MASTER.

●Características generales.

Tanto las líneas SDA como SCL son líneas bidireccionales que se conectan a +Vdd mediante resistencias de carga pull-up.

Cuando el bus está libre ambas líneas están a nivel lógico “1”. Los transistores de salida conectados a las líneas del bus I2C deben ser de colector abierto para conectar mediante conexión tipo AND.

Los bits de datos sobre el bus pueden transferirse a una velocidad de 100 Kbits/s. La capacidad máxima en el bus es de 400 pF y el número de dispositivos conectados no debe superarla.

●Transferencia del bit.

Debido a la variedad de tecnología empleada en los dispositivos diseñados para conectarse al bus I2C (CMOS, NMOS, TTL, etc.) los niveles lógicos “0” y “1” de los bits transferidos, no tienen una tensión fija si no que dependen de la tensión Vdd. Cada bit que se transfiere por la línea SDA debe ir acompañado de un pulso de reloj por la línea SCL.

-Validez del bit.

El bit de datos transferido por la línea SDA debe mantenerse estable durante el periodo en que la señal de reloj está a nivel “1”. La línea de datos SDA sólo puede cambiar de estado durante el periodo en que la señal de reloj esté a “0”.

-Condiciones de inicio (START) y parada (STOP).

Existen dos situaciones únicas que determinan el inicio y final de toda transferencia de datos entre el MASTER y el SLAVE.

Una transición de “1” a “0” (flanco descendente) en la línea SDA al tiempo que SCL está a nivel lógico “1”, determina la condición de START. Una transición de “0” a “1” (flanco ascendente) sobre la línea SDA al tiempo que SCL permanece a nivel “1” determina la condición de STOP.

Ambas condiciones son siempre generadas por el MASTER. El bus se considera ocupado (BUSY) tras la condición de START y, se considera libre después de la condición de STOP.

2.2.- La comunicación EXTERNA: El canal serie RS-232

Esta formado por el módulo SCI (Interface de Comunicación Serie), y permite la comunicación serie de la placa con cualquier periférico externo.

La aparición del canal serie en la etapa de ampliación de la Trainer nos permitirá, tal como veremos posteriormente, una comunicación con el PC utilizando este canal. Esta es, seguramente el módulo más importante de la Trainer PLUS, ya que incorpora un elemento que la Trainer no tenía y que es de vital importancia en cualquier proceso actual.

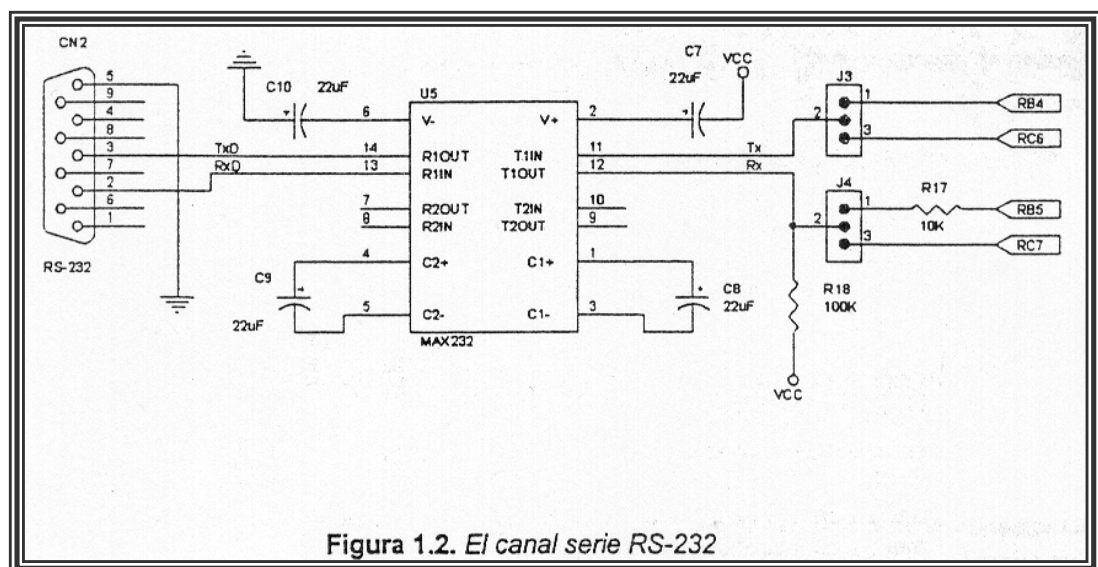
De este modo podremos probar si el procesador es capaz de gestionar y mantener la comunicación con el PC de la manera que nosotros deseamos, y poder corregir o arreglar el programa para poder llegar a los objetivos marcados.

•Funcionalidad

Está formado por el circuito MAX232 (U5). Convierte los niveles lógicos TTL presentes en la patilla 11 de transmisión (Tx), a niveles lógicos RS-232, en la patilla 14 (TxD) y se aplican al conector DB9 (CN2).

También tenemos la señal de recepción RxD que llega desde el canal serie a la patilla 13 con niveles lógicos RS-232 que son convertidos a niveles lógicos TTL obteniéndolos por la patilla 12 (Rx).

Las señales Tx y Rx se controlan por hardware desde RC6 y RC7 respectivamente, si se trabaja con un PIC que se gestione por software se utilizan RB4 y RB5. Para la selección de un otro lo haremos con los jumpers J3 y J4.



3.- CAPÍTULO III: IMPLEMENTACIÓN Y RECURSOS FÍSICOS

3.1.- Listado del material necesario.

Para poder programar el microcontrolador, que en el caso que nos ocupa es un 16F84, deberemos de tener el siguiente material.

- Un PC.
- Software adecuado: Dentro de este apartado incluiremos, tanto el compilador MPASM, como el programador PICME-TR, así como los manuales de ayuda necesarios para su correcta utilización.
- Cable de conexión serie con el PC: Para poder realizar las aplicaciones sobre la placa en el campo de comunicación, deberemos tener correctamente conectado, un cable de comunicación serie entre PC y la Trainer Plus.
- Disquettes MicroPic Trainer PLUS, con las librerías necesarias para gobernar los periféricos de la placa.
- Documentación teórica, como puede ser el manual de usuario del MicroPic Trainer PLUS, o el dossier con los distintos ejemplos a programar.
- La placa MicroPic Trainer con su ampliación (PLUS), así como también su correspondiente fuente de alimentación, y el cable conexión paralelo que nos comunique con el PC, para poder descargar datos.
- El microcontrolador a programar, así como también si se desconoce su patillaje, el Data-Sheet correspondiente.

3.2.- Discusión del software a utilizar.

En un principio nosotros y el grupo que hacia el estudio sobre la Trainer, pensábamos utilizar un programa como el MPLAB, que se mueve en entorno Windows, lo cual a priori, nos proporcionaba mayor seguridad y claridad a la hora de desenvolvemos. Pero después de la primera toma de contacto con los programas en una sesión de clase, nos dimos cuenta que el tiempo para aprender este tipo de software era mayor del que nosotros teníamos, así que decidimos utilizar un compilador auxiliar como el MPASM, y el programa de volcado PICME-TR que entrega el propio fabricante.

Tal como explicaremos en el siguiente capítulo estos dos programas se usan en entorno MS-DOS con lo cual su ejecución de trabajo resulta más lenta y pesada al principio, pero con la práctica se puede llegar a compensar.

4.- CAPÍTULO IV: PROGRAMACIÓN CON EL μ PIC-TRAINER PLUS

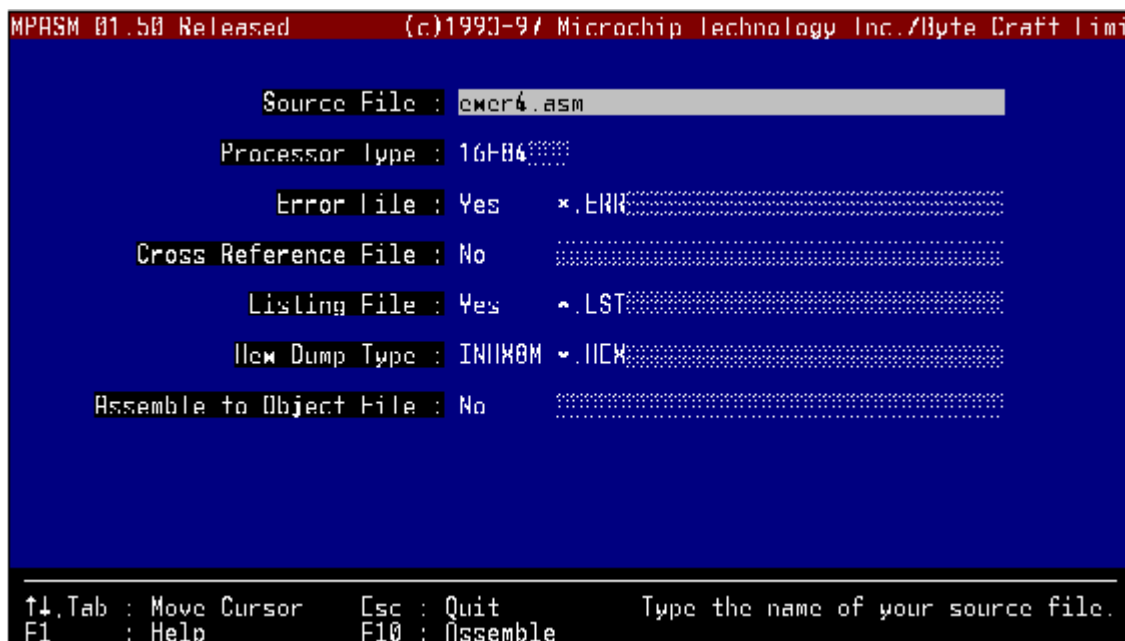
Una vez dispuestos a programar el PIC podríamos hacer una evaluación previa a través de un simulador como el SIMUPIC (pels PIC16F84 y PIC16C84). Aunque también se puede optar, como es nuestro caso, programarlo directamente a través de la Trainer.

4.1.- El ensamblado

•El MPASM

Para realizar el ensamblado de código, hemos utilizado un ensamblador de entorno DOS, que simplemente nos pasa el fichero de extensión .ASM a extensión .HEX. De tal manera que, posteriormente, se puede realizar el volcado del archivo desde el programa hasta el PIC a programar.

La pantalla de presentación del MPASM es muy sencilla. Los pasos a seguir para el correcto ensamblado de un programa se detallan a continuación.



PROCEDIMIENTO:

Antes de todo debemos de tener el código escrito en un fichero con extensión .ASM, esto se puede realizar desde el editor de MS-DOS, por ejemplo, tan solo hay que nombrar el archivo y darle extensión .ASM.

1º.- SOURCE FILE→Debemos especificar la ruta de acceso donde se encuentra el fichero a ensamblar.(.ASM)

2º.- PROCESSOR TYPE→En este apartado pondremos el tipo de procesador a programar en nuestro caso el 16F84.

3º.- ERROR FILE→Tenemos la opción de que nos cree un fichero donde nos ponga los errores que se produzcan en el proceso de ensamblaje, en el caso que los haya.

4º.- CROSS REFERENCE FILE→Crea un listado con las etiquetas encontradas.

5º.- LISTING FILE→Esta opción nos permite crear un fichero tipo .LST. Este archivo nos informará paso a paso de las incidencias del ensamblaje.

6º.- HEX DUMP TYPE→Aquí pondremos el tipo de fichero .HEX que queremos crear, en nuestro caso escogemos la opción por defecto INHX8M. Además incluiremos la ruta de acceso donde queremos que el archivo sea almacenado. Nos creará el archivo por defecto que introduciremos en la memoria del PIC.

7º.- ASSEMBLE OBJECT FILE→Si seleccionamos esta opción nos crea un fichero .OBJ.

Para el ensamblaje del programa solo es necesario seguir los pasos 1, 2 y 5, como opcional podríamos escoger que nos creara un .LST o un .ERR, si el código fuera creado por nosotros mismos y tuviéramos posibilidad de obtener errores en la ejecución del código máquina.

4.2.- La carga de datos

•El PICME-TR

Es el programa encargado de volcar la información al PIC. Su pantalla de trabajo es la siguiente:

The screenshot displays the PICME-TR v2.4B (Oct 2000) interface by MICROSYSTEMS ENGINEERING BILBAO. The main window is divided into several sections:

- BUFFER DE MEMORIA DE PROGRAMA:** A table showing memory addresses from 0000 to 0078, each with eight 3FFF values.
- SELECCION DE MODELO:** Radio buttons for 16C6X, 16C62X, 12CXXXX, 16C7XX, and 16F8XX (selected).
- COMPROBACION DE BORRADO:** Radio buttons for SI and NO (selected).
- PALABRA DE CONFIGURACION:** Four sub-sections with radio buttons:
 - Código P:** SI, NO (selected).
 - Power Timer:** SI, NO (selected).
 - Watchdog T:** SI, NO (selected).
 - Oscilador:** RC, HS, XT (selected), LP.
- MODELO ELEGIDO:** 16F84, 18 pines, 13 E/S, 1K Prg., 68 RAM, 64 Datos.
- ID/CHECK:** 0000 C596.
- Buttons:** Abrir Fichero, Programar Todo, Programar Palabra, Verificar, Leer PIC, Borrar, Comprobar Borrado, SALIR, Editar Datos, Ayuda.

Como se puede observar todas las acciones están en castellano, por lo que es de muy fácil manejo.

A continuación procederemos a realizar una breve descripción de cada una de las acciones que puede realizar:

- ABRIR FICHERO:** Pestaña que abre el fichero a volcar, solamente abre archivos con extensión .HEX.

- PROGRAMAR TODO:** Vuelca todo lo que hay en el buffer de memoria del programa hacia la memoria EEPROM/FLASH del PIC.

- VERIFICAR:** Realiza una verificación de la conexión de placa a PC. También comprueba que todos los Jumpers estén colocados en su correcta posición.

- LEER PIC:** Lee el contenido de la memoria EEPROM/FLASH del PIC y lo muestra por pantalla.

- BORRAR:** Realiza un borrado de la memoria EEPROM/FLASH del PIC siempre que éste lo permita.

- COMPROBAR BORRADO:** Comprueba el borrado realizado sobre el PIC.

- SELECCIÓN DEL MODELO:** Permite seleccionar el modelo de procesador a programar.

- PALABRA DE CONFIGURACIÓN**

- Código P: (protección del código), seleccionando esta opción se funde un fusible interno en el momento de gravar el chip, así se evita que pueda ser leído. Si desactivamos la opción el chip puede ser leído.

- Watchdog: es un temporizador, se encarga de provocar un reset y reinicializar el microcontrolador si el software de la aplicación no lo refresca cada cierto tiempo. Si seleccionamos esta opción el software del sistema se ira refrescando para evitar el reset.

- Power Timer: es un temporizador que cuando se activa el proceso de reinicialización el microcontrolador tarda unos 70 ms en realizarlo para dar tiempo a que se establezca la tensión de alimentación.

- OSCILADOR:** Nos da la posibilidad de escoger el oscilador que queramos, dentro de una gama (RC, XT, HS, LP), en la que todos trabajan a distinta frecuencia.

Los tipos de osciladores son:

- **Oscilador “RC”:**

Es un oscilador constituido por una simple resistencia y un condensador. Este oscilador de bajo coste proporciona una no buena estabilidad de la frecuencia el valor de la cual viene determinada por el valor de los componentes R-C.

- **Oscilador “HS”:**

Este logra una elevada velocidad comprendida entre 4 i 10 MHz y se basa en un cristal de cuarzo o resonador ceramico.

- **Oscilador “XT”:**

Se trata de un oscilador de cristal o resonador para frecuencias estandars comprendidas entre los 100 KHz y los 4 MHz.

- **Oscilador “LP”:**

Consiste en un oscilador de bajo consumo con cristal o resonador con un rango de frecuencias de trabajo de 35 a 200 KHz.

El cristall de quars és ubicat entre els pins OSC1 i OSC2 del PIC.

4.3.- Las librerias

El fabricante nos facilita una serie de librerias para poder programar con éxito los distintos periféricos que posee la Trainer. A continuación explicaremos las distintas librerias disponibles:

•**P16CXX.INC**: Se trata de un fichero proporcionado por Microchip que contiene la definición de todos los registros de los diferentes dispositivos media

•**TECLADO.INC**: Se trata de un fichero que incluye las rutinas encargadas de explorar y gestionar el teclado. Sus subrutinas son:

-La rutina “**KEY_SCAN**” se encarga de explorar el teclado, devuelve en la variable “Tecla el código de la tecla pulsada o bien el código 0x80 en caso de que no haya tecla pulsada.

-La rutina “**KEY_BCD**” se encarga de convertir a código BCD (de 0 a F) el código de la tecla pulsada, el resultado de la conversión se devuelve en la misma variable “Tecla”.

-La rutina “**BCD_7SEG**” convierte el código BCD presente en el registro W en su correspondiente a 7 segmentos.

•**LCD_CXX.INC**: Se trata de un fichero que incluye las rutinas encargadas de visualizar y gestionar el LCD. Sus subrutinas son:

-La rutina “**LCD_DATO**” envía el carácter ASCII contenido en el registro W al módulo LCD. La misma rutina se encarga de verificar que el módulo no esté ocupado así como de generar las señales R/W, RS y E del mismo. En definitiva se encarga de visualizar sobre el LCD.

-La rutina “**LCD_INI**” se encarga de inicializar el LCD según las especificaciones del fabricante.

-La rutina “**LCD_REG**” envía el código presente en W al módulo LCD. Este código es uno de los diferentes códigos de control con los que se pueden conseguir diferentes efectos de visualización sobre el LCD. La rutina se encarga de verificar el estado interno del módulo así como de generar las señales R/D, RS y E.

-La rutina “**UP_LCD**” se encarga de configurar los puertos A y B para el control del LCD.

•**RS232LOW.INC**: Se trata de un fichero que incluye las rutinas encargadas de controlar y facilitar la comunicación vía canal serie. Estas rutinas hacen uso de una serie de variables que deben estar cargadas desde el programa fuente que las emplee. Con estas variables se establecen los baudios, nº de bits por carácter, nº de bits de stop, etc. Sus subrutinas son:

-La rutina “**RxD**” comprueba si recibimos algún carácter y lo deposita en la variable “Rxdreg”.

-La rutina “**TxD**” transmite el carácter almacenado en la rutina “Txdreg”.

•**I2CLOW.INC**: Se trata de un fichero que incluye las rutinas que permiten que un PIC de gama media pueda controlar y gestionar la comunicación a través del bus I²C, en el modo master, mediante software. Con estas rutinas es posible controlar los 4 dispositivos I²C de que consta el sistema µPIC Trainer Plus: Controlador de display, reloj/calendario, E/S paralelo, ADC/DAC. Sus subrutinas son:

-La rutina “**Inicia_Bus**” configura las líneas RB6 y RB7 como líneas SDA y SCL del BUS I²C, dejándolas en alta impedancia que es el estado del bus en reposo.

-La rutina “**Txt_Start_Bit**” transmite la secuencia de inicio según el protocolo I²C.

-La rutina “**Txt_Byte**” transmite el byte previamente almacenado en la variable “Databyte”, al tiempo que se van generando los pulsos de reloj.

-La rutina “**Txt_Stop_Bit**” transmite la secuencia de Stop según el protocolo I²C.

-La rutina “**RcvByte**” espera recibir un byte y lo almacena en la variable “DataByte”. Si el bit _Ultimo_Byte de la variable “Bus_Control” está a 1 se considera que es el último byte a recibir y se genera por tanto el bit NACK. En caso contrario se genera el ACK.

4.4.- Aplicaciones

•Descripción del Proceso

A continuación se intentará hacer una descripción del proceso a seguir para poder programar el PIC 16F84.

1.-Escribir el código a programar en un editor. Se puede usar el editor de MS-DOS, o cualquier editor de notas utilizado en entorno Windows, teniendo presente guardarlo como fichero .ASM.

2.- Ensamblar el programa fuente utilizando el ensamblador MPASM versión MS-DOS, siguiendo los pasos dictados en el apartado “4.1.- El ensamblado” .

3.- Conectar, mediante el bus de expansión, µPIC Trainer con µPIC Trainer PLUS.

4.- Alimentar el sistema y comprobar el estado de los jumpers, tanto los de µPIC Trainer como los de µPIC Trainer PLUS. Teniendo en cuenta que las señales que se emplean para la

lectura/grabación de un PIC son RB6 y RB7 y éstas a su vez están conectadas a los leds, display y módulo LCD. Por lo tanto es importante desconectar estos periféricos mediante los jumpers J7, J5 y J6 (ubicados en la μ PIC Trainer), respectivamente, para que no supongan carga alguna a dichas señales y alteren la información leída o grabada.

Además se deberá comprobar el estado de los jumpers J1, J2, J3 y J4 (ubicados en la μ PIC Trainer PLUS) teniendo en cuenta la siguiente tabla:

JUMPER	2-1	2-3	DESCRIPCIÓN
J1-J2	Abierto	Abierto	Bus I ² C desconectado
J1-J2	Abierto	Cerrado	El Bus I ² C está controlado mediante RC3 y RC4. Se emplea en aquellos PIC's que tengan I ² C implementado por hardware
J1-J2	Cerrado	Abierto	El Bus I ² C está controlado mediante RB6 y RB7. Se emplea en ocasiones en las que el bus se gestiona por software
J3-J4	Abierto	Abierto	Canal serie RS-232 desconectado
J3-J4	Abierto	Cerrado	La transmisión se realiza por RC6 y la recepción por RC7. Se emplea cuando los PIC's tengan la USART implementada por hardware.
J3-J4	Cerrado	Abierto	La transmisión se realiza por RB4 y la recepción por RB5. Se emplea cuando el canal serie se gestiona por software

5.-Una vez comprobado el estado de los jumpers y realizado el ensamblaje tendremos un fichero .HEX.

Desde el programa PICME-TR, abrir dicho fichero y cargarlo en el PIC, tal como se indica en el apartado “4.2.-La carga de datos”.

6.- Si no ha habido ningún error de verificación, y el PIC estaba borrado y en perfecto funcionamiento, el PICME-TR nos informará que el PIC ha sido programado, y podremos comprobar los resultados obtenidos.

●Enlace entre arquitectura y aplicaciones

Para poner en práctica todos los conocimientos adquiridos se han realizado ciertas aplicaciones. Además, así se puede comprobar la evolución de la μ PIC Trainer PLUS a medida que éstas se desarrollan. También podremos observar lo anunciado teóricamente en los capítulos anteriores en cuanto a su arquitectura.

Para programar el PIC se deben tener en cuenta todos los pasos anteriormente citados en el apartado de procedimiento.

APLICACIÓN 1

La primera aplicación tratará de explicar como es posible gobernar dos periféricos distintos conectados a la misma puerta (LCD y Teclado). Para ello basta reprogramarla dinámicamente en el tiempo según sea necesario.

El listado del código sería el siguiente:

```

List    p=16F84           ;Tipo de procesador
include"P16F84.INC"       ;Definiciones de registros internos

Lcd_var      equ    0x0c      ;Inicio de las variables para el LCD
Key_var      equ    0x0e      ;Inicio de las variables del teclado

Temporal_1   equ    0x13      ;Variable temporal nº 1
Temporal_2   equ    0x14      ;Variable temporal nº 2

                org    0x00      ;Vector de reset
                goto   Inicio
                org    0x05      ;Salva vector de interrupción

                include"LCD_CXX.INC" ;Incluir rutinas de manejo del LCD
                include"TECLADO.INC" ;Incluir rutinas de manejo del teclado

Tabla_Mensajes:    movwf PCL      ;Desplazamiento sobre la tabla

Mens_0          equ    $      ;Mens_0 apunta al primer carácter
                dt      "Se ha pulsado: ",0x00

;Mensaje: Esta rutina visualiza en el LCD el mensaje cuyo inicio está indicado en
;el acumulador. El fin de un mensaje se determina mediante el código 0x00

Mensaje        movwf Temporal_1 ;Salva posición de la tabla
Mensaje_1      movf  Temporal_1,W ;Recupera posición de la tabla
                call  Tabla_Mensajes ;Busca caracter de salida
                movwf Temporal_2 ;Guarda el caracter
                movf  Temporal_2,F
                btfs STATUS,Z ;Mira si es el último
                goto  No_es_ultimo
                return
No_es_ultimo    call  LCD_DATO ;Visualiza en el LCD
                incf  Temporal_1,F ;Siguiente caracter
                goto  Mensaje_1

Inicio:        bsf    STATUS,RP0 ;Selecciona página 1 de datos
                movlw b'00001111'
                movwf OPTION_REG ;Preescalr de 128 asignado al WDT
                bcf    INTCON,GIE ;Desconectar interrupciones
                bcf    STATUS,RP0 ;Selecciona página 0 de datos
                clrwdt ;Refrescar WDT

                call  UP_LCD ;Configura Puerta A y B como salidas
                call  LCD_INI ;Rutina de inicialización del LCD
                movlw b'00001100'
                call  LCD_REG ;LCD en ON
                movlw b'00000001'
                call  LCD_REG ;Borra LCD y HOME

```

;Salida del mensaje "Tecla pulsada:"

```
    movlw Mens_0
    call  Mensaje           ;Visualiza el mensaje
```

;Bucle principal

```
Loop:    clrwdt             ;Refrescar el WDT
         call  Key_Scan     ;Explora el teclado
         movlw 0x80
         subwf Tecla,W
         btfsc STATUS,Z     ;Hay alguna pulsada ?
         goto  Loop        ;No, seguir explorando

         call  Key_BCD      ;Convierte código de tecla a código BCD
         movlw 0x8f
         call  LCD_REG      ;Posiciona el cursor del LCD
         movf  Tecla,W
         sublw .9
         btfss STATUS,C     ;Es mayor que 9 (A, B,C,D,E,F)?
         goto  Mayor_que_9 ;Si
         movf  Tecla,W      ;No
         addlw 0x30         ;Ajuste ASCII de los caracteres del 0 al 9
         call  LCD_DATO     ;Visualizar sobre el LCD
         goto  Loop

Mayor_que_9 movf Tecla,W
         addlw 0x37         ;Ajuste ASCII de los caracteres de la A a la F
         call  LCD_DATO     ;Visualiza sobre el LCD
         goto  Loop

end      ;Fin del programa fuente
```

APLICACIÓN 2

En esta aplicación, trabajaremos con el canal serie, para ello necesitaremos un equipo terminal o similar conectado al canal RS-232, con tal de poder realizar las pruebas de transmisión y recepción oportunas. Nosotros hemos utilizado el canal serie COMX de un PC.

Antes de hacer nada es importante indicar que se debe ejecutar el fichero TERMINAL.EXE para convertir el PC en un emulador del canal serie y proporcionarnos la posibilidad de comunicar-nos con el.

Deberemos usar las rutinas incluidas en el fichero RS232LOW.INC, además el usuario deberá preparar los cables de comunicación entre la Trainer PLUS y el PC, teniendo en cuenta de cruzar la señal de Transmisión (TxD) con la de recepción (RxD) y viceversa.

En el ejemplo que se propone esperamos a recibir un carácter desde el terminal para, inmediatamente, retransmitirlo al mismo a modo de ECO.

El listado del programa seria el siguiente:

```
List    p=16F84          ;Tipo de procesador
        include"P16F84.INC"      ;Definiciones de registros internos

CLKIN          equ    .4000000    ;Frecuencia de trabajo
BAUDIOS        equ    .9600       ;Velocidad de comunicación
T_MODO         equ    1           ;Transmite primero bit LSB
R_MODO         equ    1           ;Recibir primero bit LSB
T_Nbit         equ    8           ;Transmite caracteres de 8 bits
R_Nbit         equ    8           ;Recibe caracteres de 8 bits
Sbit          equ    2           ;Transmite 2 bits de stop

Rs232_var     equ    0x0c        ;Inicio de las variables para rutinas RS232

        org    0x00             ;Vector de reset
        goto   Inicio
        org    0x05             ;Salvar el vector de interrupción

        include"RS232LOW.INC"    ;Incluir rutinas de comunicación

;Programa Principal

Inicio:      bsf    STATUS,RP0;Selecciona página 1 de datos
            movlw  b'10001111' ;Preescaler de 128 para el WDT. Pull_Up Off
            movwf  OPTION_REG
            movlw  b'11101111'
            movwf  TRISB        ;RB5 entrada y RB4 salida
            bcf    STATUS,RP0;Selecciona página 0 de datos

Loop        clrwdt
            bsf    Txd_pin      ;Línea de transmisión a "1"
            bsf    Rxd_pin      ;Línea de recepción a "1"

            call   RxD          ;Espera a recibir un carácter
            movf   Rxdreg,W      ;Lee el carácter recibido
            movwf  Txdreg
            call   TxD          ;Retransmite carácter recibido (ECO)

            goto   Loop

            end                ;Fin del programa fuente
```

APLICACIÓN 3

En la tercera aplicación vamos a trabajar con el dispositivo I²C y el módulo de visualización. Debemos tener en cuenta que a base de hipotecar 2 líneas del PIC para la gestión del bus obtenemos a cambio una expansión de otras 8 líneas suministradas por este dispositivo.

Deberemos pensar a la hora de conectar los jumper J1, J2, J3 y J4, que la Interface I2C va gestionada por software. Igualmente es importante desconectar los leds de la uPIC Trainer mediante el jumper J7 pues de lo contrario actuarían como carga adicional en las líneas RB6 y RB7 que, en estos ejemplos, forman el bus I2C.

Haciendo uso de las rutinas incluidas en el fichero I2CLOW.INC, se trata de controlar el funcionamiento de los distintos dispositivos I2C, existentes en la tarjeta Trainer PLUS. Estos dispositivos están conectados entre sí al bus I2C común. Se recomienda leer las especificaciones técnicas de los mismos, para tener un mayor conocimiento de las posibilidades de los mismos.

Se trata de un simple programa visualiza el valor "1234" sobre los cuatro displays de la uPIC Trainer PLUS. El valor a visualizar podría ser cualquier otro.

```
List    p=16F84           ;Tipo de procesador
include"P16F84.INC"      ;Definiciones de registros internos

CLKIN          equ    .4000000    ;Frecuencia de trabajo

I2c_var        equ    0x0c        ;Inicio de las variables para rutinas I2C_LOW
Key_var        equ    0x12        ;Inicio de las variables de manejo del teclado
Digito_1       equ    0x18
Digito_2       equ    0x19
Digito_3       equ    0x1a
Digito_4       equ    0x1b        ;Buffer de salida al SAA1064 de los 4 dígitos

org    0x00            ;Vector de reset
goto    Inicio
org    0x05            ;Salvar el vector de interrupción

include"I2CLOW.INC"      ;Incluye las rutinas de manejo del bus I2C
include "teclado.INC"    ;Incluye rutinas de manejo del teclado
```

;VISUALIZA: Esta rutina recoge el valor de las cuatro variables "Dígito" y las transmite por el bus I2C para su visualización sobre los displays de uPIC Trainer Plus.

```
VISUALIZA: call    Inicia_Bus    ;Inicia líneas del bus I2C
            call    Txt_Start_Bit ;Envía condición de inicio
            movlw b'011110110'
            movwf DataByte
            call    Txt_Byte      ;Transmite dirección del dispositivo (escritura)
            clrf    DataByte
            call    Txt_Byte      ;Transmite byte de instrucción
            movlw b'01000111'
            movwf DataByte
            call    Txt_Byte      ;Transmite byte de control
            movf    Digito_1,W
            movwf DataByte
            call    Txt_Byte      ;Transmite Dígito 1
            movf    Digito_2,W
```



```

movwfDataByte
call  Txt_Byte      ;Transmite Dígito 2
movf  Digito_3,W
movwfDataByte
call  Txt_Byte      ;Transmite Dígito 3
movf  Digito_4,W
movwfDataByte
call  Txt_Byte      ;Transmite Dígito 4
call  Txt_Stop_Bit ;Transmite secuencia de STOP
return

```

```

Inicio:  clrf  PORTA
         clrf  PORTB
         bsf   STATUS,RP0;Selecciona página 1 de datos
         clrf  TRISA
         clrf  TRISB      ;Puerta A y B salidas
         movlw b'10001111' ;Preescaler de 128 para el WDT. Pull_Up Off
         movwfOPTION_REG
         bcf   STATUS,RP0;Selecciona página 0 de datos

         call  Inicia_Bus  ;Inicia el bus I2C

         movlw 0x01        ;Dígito a visualizar (1)
         call  BCD_7seg     ;Convierte el dígito en código 7 segmentos
         movwfDigito_1     ;Almacena en el buffer de salida
         movlw 0x02        ;Dígito a visualizar (2)
         call  BCD_7seg     ;Convierte el dígito en código 7 segmentos
         movwfDigito_2     ;Almacena en el buffer de salida
         movlw 0x03        ;Dígito a visualizar (3)
         call  BCD_7seg     ;Convierte el dígito en código 7 segmentos
         movwfDigito_3     ;Almacena en el buffer de salida
         movlw 0x04        ;Dígito a visualizar (4)
         call  BCD_7seg     ;Convierte el dígito en código 7 segmentos
         movwfDigito_4     ;Almacena en el buffer de salida

         call  VISUALIZA    ;Visualiza el Buffer sobre los 4 dígitos del SAA1064

Loop     sleep              ;Modo standby
         goto  Loop

         end                ;Fin del programa fuente

```

5.- BIBLIOGRAFÍA

- **“Microcontroladores PIC. Diseño Práctico de Aplicaciones”**

Angulo J.Mª y Angulo Ignacio. McGrawHill 1997

- **“Manuales de usuario de Micro’PIC Trainer y de las tarjetas de prototipo PIC 18”**

Microsystems Engineering

- **“Micro’Pic Trainer”**

Angulo J.M. y Angulo I. Revista Española de Electrónica-Septiembre de 1997

- **“Microchip WebSite”**

<http://www.microchip.com>